pgeon applied to Overcooked-AI to explain agents' behaviour

Demonstration Track

Adrian Tormos
Barcelona Supercomputing Center
Barcelona, Spain
adrian.tormos@bsc.es

Victor Gimenez-Abalos
Barcelona Supercomputing Center
Barcelona, Spain
victor.gimenez@bsc.es

Javier Vázquez-Salceda
Universitat Politècnica de Catalunya
Barcelona, Spain
jvazquez@cs.upc.edu

Sergio Alvarez-Napagao
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center
Barcelona, Spain
sergio.alvarez@bsc.es

ABSTRACT

Policy Graphs (PGs) are a method for representing the behaviour of opaque agents by observing them in the environment and producing graphs where the state and action spaces are discretised into predicates. We present *pgeon*, a Python library that demonstrates the effectiveness of PGs in providing explanations for the behaviour of agents and we showcase it by applying it to a multi-agent cooperative environment: *Overcooked-AI*. This library illustrates how PGs can create transparent and explainable surrogate agents that closely mimic the behavior of the original agents. These features can help improving trust in environments where humans and AI systems collaborate by improving the explainability of all agents, even opaque or human.

KEYWORDS

Explainability; Multi-Agent Systems; Reinforcement Learning

ACM Reference Format:
Adrian Tormos, Victor Gimenez-Abalos, Javier Vázquez-Salceda, and Sergio Alvarez-Napagao. 2024. pgeon applied to Overcooked-AI to explain agents' behaviour: Demonstration Track. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION AND BACKGROUND

The rising adoption of opaque models for complex decision making in virtual or physical agents, such as those trained with reinforcement learning (RL), brings about ethical issues concerning the clarity and transparency on their behaviour. This is especially relevant in contexts where auditable decision-making is crucial for effective collaboration and interaction in socio-technical systems, so it is becoming increasingly important to be able to offer human-understandable explanations of the behaviour of agents, regardless of whether the agents are transparent by design or opaque. Here, we showcase *pgeon*, a practical implementation using policy graphs to tackle this issue along with a visual interface, demonstrated for the first time in a multi-agent collaborative environment. We also contribute with a novel development library for managing PGs.

Background. Krajna et al. [7] provide an exhaustive look at the challenges and technical solutions and methods in the topic of Explainability in Reinforcement Learning (XRL). There are several ways to categorise the possible explanations that can be produced from the behaviour of agents, such as how much detail they provide versus how efficient (or compact) they are; or whether they are offered after the agent has acted in the environment (post-hoc) or prior to the decision making process actually taking place (intrinsic). Other factors such as the characteristics of environment or the policy, or the number of agents acting in the environment add to the complexity of generating explanations.

From the many methods available [7], our research focuses on Policy Graphs (PGs) for understanding agent behaviour, regardless of whether the agent is transparent or opaque. A PG is formalised as a directed graph that maps states to nodes and actions to edges, where these states and actions are discrete reductions of the (potentially continuous) actual state and action spaces, usually in the form of predicates. As has been shown in previous work [1, 4, 5, 8, 10], a well-designed PG, with a correctly chosen set of predicates, can closely approximate an opaque agent's behaviour. PGs are useful in both single-agent and multi-agent environments by providing post-hoc, proactive, and global explanations.

There are other methods that use predicates for behavioural explanations of opaque agents, such as those that represent behaviour as sequences of plans or operators [2, 11, 12]. These are good for agents with clear sequential decision making goal-directed processes. PGs do not make assumptions about the agents' internal model, making them suitable for agents with concurrent goals or non-optimal decision-making processes, such as human-like agents, as already studied in [10].

2 POLICY GRAPHS FOR EXPLAINABILITY

From behavior to policy graphs. The construction of a PG begins with observing an agent's interactions within the environment, systematically recording state transitions and subsequent actions. To render an agent's behavior into a coherent PG, we reduce the observations and actions (which might be continuous) into discrete

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

states and actions, which are subsequently mapped into nodes and edges in the graph. Then, the PG is created through a frequentist approach, counting each time a transition between two states when the agent performed a certain action, and computing transition probabilities. This includes both the probability of performing an action in a state and the probability of arriving at a state when doing so. Each non-zero probability becomes an edge of the graph.

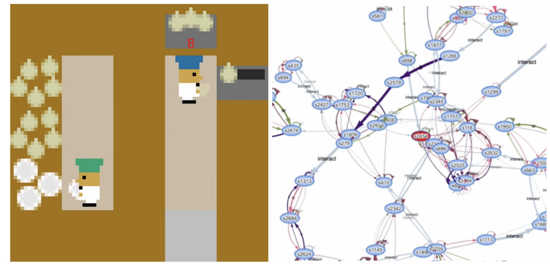
Producing explanations from policy graphs. Generating understandable explanations of agent behavior through PGs involves traversing the directed graph to illustrate the decision-making process. There are three types of questions that can be answered, and therefore there are three different algorithms that define exactly how this traversal process is executed: 1) *What will you do when you are in state X?*, 2) *When do you perform action A?*, and 3) *Why did you not perform action A in state X?*. These three algorithms produce explanations that can be converted into natural language by means of predicates, and are part of the demonstration.

Generating PG-based agents. One way to evaluate the quality of the explanation is by trying to understand whether these explanations really represent the behavior of the agent being analyzed. This can be achieved by generating an agent (called *PG agent*) that uses the graph to decide the next action based on the node of the policy graph that more closely resembles the current state and checking the probabilities of the edges representing the next actions. During operation, when the PG agent encounters a state, it queries the PG to determine the next action, adhering to the policy depicted by the graph. The agent continues to navigate through the environment, at each step referencing the PG to dictate its actions. The accuracy with which PG-based agents emulate the behavior of the original, potentially more opaque agent is vital, as it verifies the PG's reliability as a representative model.

3 PRACTICAL DEMONSTRATION

In the context of Overcooked-AI, our exploration mainly focuses on understanding agent behaviour in the presence of other agents and the dynamics of cooperation in a structured, yet complex environment. The game, inspired by the popular video game Overcooked, involves two players who must work together to prepare and serve soups. Players navigate through the kitchen, preparing ingredients, cooking, and delivering finished dishes while managing spatial constraints and task distribution.

Our demonstration primarily illustrates the process of mapping their behaviour onto PGs by means of a set of 10 predicates, two of them being related to the state of the other agent. These two predicates allow us to achieve explanations that involve the emergence of cooperation between the agents. Therefore, an interesting insight of this demonstration is that PGs can approximate the behavior of opaque agents if the set of predicates chosen is sufficiently expressive, an intelligible textual or visual representation of agent strategies and decisions in various game configurations. We include simulations of the game in which PG agents that mimic the original (PPO [9] and GAIL [6]) agents. We allow for two different policy generation strategies –Greedy and Stochastic [10]– and elucidate how they affect the performance of the agent depending on the specific layout and the need for cooperation in each one.



Current state = *{held(Nothing), pot_state(Pot0;Cooking), pot_state(Pot1;Preparing), onion_pos(Stay), tomato_pos(Stay), soup_pos(Stay), dish_pos(Stay), service_pos(Bottom), pot_pos(Pot0;Top), pot_pos(Pot1;Interact)}*

Figure 1: *pgeon* web interface, showing a running agent with its PG. The node in red is the current discretized state the blue agent is in; the edges represent the combination of actions the agent can undertake and the possible resulting states. The weight of the edge depends on the probability.

The pgeon library. We are developing a library implementing this explainability pipeline with the intention to offer a flexible framework to allow developers to generate and utilize policy graphs. It is publicly and openly released at https://github.com/HPAI-BSC/pgeon-xai. The core features of this library are:

- PG generation via Gym/PettingZoo or CSV traces.
- Surrogate policy generation based on policy graphs.
- Policy graph visualization.
- Application of all the stated query algorithms.

4 CONCLUSIONS

The demonstration aims to present a practical application and visual example of how Policy Graphs (PGs) can provide explanations for the behaviour of opaque agents in a multi-agent cooperative environment, showcasing:

- The construction and utilisation of PGs to map and represent agent behaviour in discrete states and actions.
- How PGs can be utilised to approximate the behaviour of an opaque agent into what we call a PG-based agent, providing insights into the agent's decision making.
- Analysis of different PG generation strategies and how they correlate with surrogate agent behaviour.
- A side-by-side comparison of agent behaviours with their respective PG representations, enabling viewers to observe their correlation and validate the efficacy of the PG model.

The intent of this demonstration is to offer a tangible and accessible means for viewers to explore explainability with policy graphs, without requiring advanced knowledge on behaviour modelling and PG construction. For more information about the formalisation and a discussion on the limitations of the approach, refer to [3].

A video demonstrating all the stated features of *pgeon* can be found at the URL https://vimeo.com/tranchis/pgeon.

ACKNOWLEDGMENTS

This work has been partially supported by the AI4CCAM (Grant agreement ID: 101076911) and the HumanE-AI-Net (Grant agreement ID: 952026) European projects.

REFERENCES

[1] Antoni Climent, Dmitry Gnatyshak, and Sergio Alvarez-Napagao. 2021. Applying and Verifying an Explainability Method Based on Policy Graphs in the Context of Reinforcement Learning. In *Frontiers in Artificial Intelligence and Applications*, Mateu Villaret, Teresa Alsinet, Cèsar Fernández, and Aïda Valls (Eds.). IOS Press. https://doi.org/10.3233/FAIA210166

[2] Devleena Das, Sonia Chernova, and Been Kim. 2023. State2Explanation: Concept-Based Explanations to Benefit Agent Learning and User Understanding. https://openreview.net/forum?id=xGz0wAIJrS

[3] Marc Domenech i Vila, Dmitry Gnatyshak, Adrian Tormos, Victor Gimenez-Abalos, and Sergio Alvarez-Napagao. 2024. Explaining the Behaviour of Reinforcement Learning Agents in a Multi-Agent Cooperative Environment Using Policy Graphs. *Electronics* 13, 3 (2024), 573.

[4] Marc Domènech i Vila, Dmitry Gnatyshak, Adrián Tormos, and Sergio Alvarez-Napagao. 2022. Testing Reinforcement Learning Explainability Methods in a Multi-Agent Cooperative Environment. In *Frontiers in Artificial Intelligence and Applications*, Atia Cortés, Francisco Grimaldo, and Tommaso Flaminio (Eds.). IOS Press. https://doi.org/10.3233/FAIA220358

[5] Bradley Hayes and Julie A. Shah. 2017. Improving Robot Controller Transparency Through Autonomous Policy Explanation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*. ACM, Vienna Austria, 303–312. https://doi.org/10.1145/2909824.3020233

[6] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. *Proceedings of the 30th International Conference on Neural Information Processing Systems* (2016), 4572–4580.

[7] Agneza Krajna, Mario Brcic, Tomislav Lipic, and Juraj Doncevic. 2022. Explainability in reinforcement learning: perspective and position. *arXiv preprint arXiv:2203.11547* (2022).

[8] Tongtong Liu, Joe McCalmon, Thai Le, Md Asifur Rahman, Dongwon Lee, and Sarra Alqahtani. 2023. A novel policy-graph approach with natural language and counterfactual abstractions for explaining reinforcement learning agents. *Autonomous Agents and Multi-Agent Systems* 37, 2 (Aug. 2023), 34. https://doi.org/10.1007/s10458-023-09615-8

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. https://doi.org/10.48550/arXiv.1707.06347 arXiv:1707.06347 [cs].

[10] Adrián Tormos, Víctor Gimenez-Abalos, Marc Domènech i Vila, Dmitry Gnatyshak, Sergio Alvarez-Napagao, and Javier Vázquez-Salceda. 2023. Explainable agents adapt to human behaviour. In *Proceedings of the First International Workshop on Citizen-Centric Multi-Agent Systems (CMAS'23)*. 42–48. https://upcommons.upc.edu/handle/2117/390757

[11] Pulkit Verma, Rushang Karia, and Siddharth Srivastava. 2023. Autonomous Capability Assessment of Sequential Decision-Making Systems in Stochastic Settings (Extended Version). In *Proceedings of the International Conference on Automated Planning and Scheduling*, Vol. 33 (1). arXiv, 189–197. https://doi.org/10.48550/arXiv.2306.04806 arXiv:2306.04806 [cs].

[12] Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. 2022. Discovering User-Interpretable Capabilities of Black-Box Planning Agents. arXiv. https://doi.org/10.48550/arXiv.2107.13668 arXiv:2107.13668 [cs].